Experiment: 8

Name – Rahul Kumar

UID – 20BCS7081

Section – 17/B

Date – 14 Nov 2023

Course – Computer vision Lab

**Aim:** Write a program to determine the effectiveness of incorporating optical flow analysis into object tracking algorithms.

**Software Required:** Matlab

# Description:

* **Data Preparation:** Start by preparing your image dataset, which includes loading and preprocessing the data. Ensure that you have a labeled dataset with images and corresponding labels for supervised learning.
* **Model Selection:** Choose a set of classification models you want to evaluate. These models can include traditional machine learning algorithms like Support Vector Machines (SVM), Random Forest, and K- Nearest Neighbors (KNN), as well as deep learning models like Convolutional Neural Networks (CNNs).
* **Training and Evaluation:** Split your dataset into training and testing sets to train and evaluate each selected model. Train each model on the training data and evaluate its performance on the testing data using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrices.
* **Cross-Validation (Optional):** To ensure robustness and reduce the risk of overfitting, consider using k- fold cross-validation. This technique involves splitting the data into k subsets and training/evaluating the model k times, using a different subset as the test set each time.
* **Comparison and Selection:** After evaluating all the models, compare their performance metrics to determine which model performs best for your image recognition task. Select the model that achieves the highest accuracy or the most suitable metric based on your application's requirements.

# Algorithm:

**Step 1:** Import Libraries

**Step 2:** Load and Preprocess Data

**Step 3:** Split Data into Training and Testing Sets **Step 4:** Define a List of Classification Models **Step 5:** Initialize Evaluation Metrics Containers **Step 6:** Train and Evaluate Models

**Step 7:** Compare Model Performance

**UID:** 20BCS7081

**Step 8:** Visualize Results

**Step 9:** Fine-tune and Optimize

* **CODE**

load RRSignsGroundTruth.mat

objectTrainingData=objectDetectorTrainingData(gTruthTrain); acfDetector=trainACFObjectDetector(objectTrainingData); imdsTest=imageDatastore("Codes Matlab\Computer Vision for Engineering and Science\Data\MathWorks Images\Railroad Signs\Test\"); bboxes=detect(acfDetector,imdsTest);

for idx=1:height(imdsTest.Files)

img=readimage(imdsTest,idx);

imgAnn=insertObjectAnnotation(img,"rectangle",bboxes.Boxes{idx}, ...

"Score:"+bboxes.Scores{idx},"FontSize",40,"LineWidth",10); figure imshow(imgAnn) end

**OUTPUT=**











